

Appln No. 09/478,682

Amdt date January 21, 2004

Reply to Office action of October 27, 2003

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

---

1. (Currently Amended) A method for testing a computer program comprising the steps of:

parsing a source code of the computer program to identify behavior of functions in the source code;

responsive to the identified behavior of functions, generating stubs to be called by respective functions in the source code for testing the source code;

instrumenting the parsed source code with the generated stubs;

compiling the instrumented code;

testing the compiled code; and

reporting test results.

2. (Previously Presented) The method of claim 1 wherein, the step of generating stubs comprises generating source code for replacing the name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

3. (Original) The method of claim 2 wherein, the specific functions are one or more of predetermined functions and predetermined stubs.

**Appln No. 09/478,682**

**Amdt date January 21, 2004**

**Reply to Office action of October 27, 2003**

4. (Original) The method of claim 3 wherein, the predetermined functions and stubs are automatically generated.

5. (Original) The method of claim 4 wherein, the step of automatically generating functions and stubs comprises automatically generating arguments to the functions and automatically initializing class members.

6. (Original) The method of claim 2 wherein, the specific functions are user-specified functions.

7. (Previously Presented) The method of claim 6 wherein, the user-specified functions are specified within a GUI.

8. (Original) The method of claim 1 further comprising the steps of breaking down the computer program into smaller components before compiling and testing the smaller components individually.

9. (Original) The method of claim 8 wherein, the smaller components are of the type of one or more of name space, class, function, and objects.

10. (Previously Presented) The method of claim 1 wherein, the step of generating stubs comprises reconstructing a class by removing the source code that is not related to the class.

11. (Previously Presented) The method of claim 1 wherein, the step of generating stubs comprises reconstructing

**Appln No. 09/478,682**

**Amdt date January 21, 2004**

**Reply to Office action of October 27, 2003**

a class by ignoring the source code that is not related to the class.

12. (Original) The method of claim 2 further comprising maintaining a list of related functions to be replaced for each function under test.

13. (Original) The method of claim 1 further comprising monitoring test coverage of the computer program.

14. (Original) The method of claim 13 further comprising displaying the monitored test coverage in a GUI as the test progresses.

15. (Original) The method of claim 1 further comprising the steps of defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined information as a separate object; and linking the compiled object to the code.

16. (Original) The method of claim 1 wherein, the step of testing comprises of white-box testing.

17. (Original) The method of claim 1 wherein, the step of testing comprises of black-box testing.

18. (Original) The method of claim 1 wherein, the step of testing comprises of regression testing.

19. (Currently Amended) A method for testing a computer program having a source code comprising the steps of:

Appln No. 09/478,682

Amdt date January 21, 2004

Reply to Office action of October 27, 2003

parsing the source code of the computer program to identify behavior of a plurality of smaller components in the source code;

breaking down the source code into the plurality of smaller components;

based on the identified behavior of plurality of smaller components, generating stubs to ~~replace~~ emulate some of the identified behavior of plurality of smaller components;

testing the plurality of smaller components individually with the generated stubs emulating some of the identified behavior; and

reporting test results.

20. (Original) The method of claim 19 wherein, the smaller components are of the type of one or more of name space, class, function, and objects.

21. (Original) The method of claim 19 further comprising replacing the name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

22. (Original) The method of claim 21 wherein, the specific functions are one or more of predetermined functions and predetermined stubs.

23. (Original) The method of claim 22 wherein, the predetermined functions and stubs are automatically generated.

**Appln No. 09/478,682**

**Amdt date January 21, 2004**

**Reply to Office action of October 27, 2003**

24. (Original) The method of claim 23 wherein, the automatically generating functions and stubs comprises automatically generating arguments to the functions and automatically initializing class members.

25. (Original) The method of claim 21 wherein, the specific functions are user-specified functions.

26. (Original) The method of claim 19 further comprising monitoring test coverage of the computer program.

27. (Previously Presented) The method of claim 26 further comprising displaying the monitored test coverage in a GUI as the test progresses.

28. (Original) The method of claim 19 further comprising the steps of defining a specific behavior when a function within the source code is called; storing the defined information; compiling the defined information as a separate object; and linking the compiled object to the code.

29. (Original) The method of claim 19 wherein, the step of testing comprises of white-box testing.

30. (Original) The method of claim 19 wherein, the step of testing comprises of black-box testing.

31. (Original) The method of claim 19 wherein, the step of testing comprises of regression testing.

32. (Currently Amended) . A system for testing a computer program comprising:

Appln No. 09/478,682

Amdt date January 21, 2004

Reply to Office action of October 27, 2003

means for parsing a source code of the computer program to identify behavior of functions in the source code;

means for generating stubs to be called by respective functions in the source code for testing the source code responsive to the identified behavior of functions;

means for instrumenting the parsed source code with the generated stubs;

means for compiling the instrumented code;

means for testing the compiled code; and

means for reporting test results.

33. (Previously Presented) The system of claim 32 wherein, the means for generating stubs comprises generating source code for replacing the name of externally called functions within the source code with the name of specific functions with same signature as the externally called functions.

34. (Original) The system of claim 32 further comprising means for breaking down the computer program into smaller components before compiling and means for testing the smaller components individually.

35. (Original) The system of claim 32 further comprising means for monitoring test coverage of the computer program.

36. (Original) The system of claim 32 further comprising means for defining a specific behavior when a function within the source code is called; means for storing the defined information; means for compiling the defined information as a

Appln No. 09/478,682

Amdt date January 21, 2004

Reply to Office action of October 27, 2003

separate object; and means for linking the compiled object to the code.

37. (Currently Amended) A computer readable medium having stored thereon a set of instructions including instruction for testing a computer program, the instructions, when executed by a computer, cause the computer to perform the steps of:

1  
B  
parsing a source code of the computer program to identify behavior of functions in the source code;

generating stubs to be called by respective functions in the source code for testing the source code responsive to the identified behavior of functions;

instrumenting the parsed source code with the generated stubs;

compiling the instrumented code;

testing the compiled code; and

reporting test results.

---